



software MS-DOS

a cura di Valter Di Dio

Eccoci così alla seconda puntata del software MS-DOS, naturalmente non si è ancora attivato il feedback con i lettori, infatti questo è il numero di luglio-agosto e, quindi, molti di voi se lo staranno godendo su una bella spiaggia assolata, mentre per me che scrivo, è una cupa e piovosa domenica di giugno. Tutto questo per spiegare che in pratica la rubrica vera e propria inizierà quando cominceranno ad arrivare le prime reazioni, i primi consigli e le prime (spero poche) lamentele.

Le lettere che ci inviate sono infatti per noi indispensabili per poter correggere il tiro e fare in modo che sia sempre il più possibile la vostra rivista; aspetteremo quindi con ansia e con piacere i primi consigli e i primi lavori che ci vorrete inviare.

Intanto eccovi le prime fatiche che alcuni lettori avevano inizialmente inviato ad altre rubriche (e che ho immediatamente «razzato»), ed un po' di posta che riguardava l'MS-DOS scippata direttamente all'arrivo.

Accesso a nuovi comandi

di Felice Sobrero - Torino

Chiamando l'interrupt 21h è possibile accedere ad alcune interessanti funzioni del Dos che non sono disponibili normalmente da tastiera. È naturalmente possibile effettuare delle chiamate a questo interrupt da programma, soprattutto «C» e Pascal offrono delle facilitazioni a questo ri-

guardo. Tuttavia, alcune di queste funzioni è utile averle a portata di mano. Vedremo come creare dei file. COM che andranno ad aggiungersi ai vari comandi esterni dell'MS-DOS già presenti sul nostro dischetto di sistema.

MOVDIR

Sembra strano, ma in MS-DOS non c'è un modo semplice e rapido per spostare un file da una directory ad un'altra nell'ambito di uno stesso disco. Normalmente l'unica cosa da fare è copiarlo e poi cancellare il vecchio, sempre che ci sia spazio di parcheggio

sufficiente e rischiando inutilmente degli errori di trascrizione.

Ma una soluzione esiste. A partire dalla versione 2.00 del Dos è disponibile tramite l'interrupt 21h la funzione numero 56h, che è in fondo una generalizzazione del RENAME: permette di cambiare non solo il NOME ma anche il PERCORSO (o path) che identifica un file. Significa che potremo cambiare il nome di un file, o la directory di appartenenza, o entrambi. Problemi di spazio non ce ne sono, perché il contenuto del file in effetti non si sposta.

La funzione, che ho battezzato

MOVDIR. BAS

```
100 OPEN "movdir.com" AS #1 LEN = 1
110 FIELD #1,1 AS AS
120 CHECKSUM = 0 : RESTORE
130 READ BYTE%
140 IF BYTE% = -1 THEN 170
150 LSET AS = CHR$(BYTE%) : PUT #1
160 CHECKSUM = CHECKSUM + BYTE% : GOTO 130
170 READ CONTROL : IF CONTROL <> CHECKSUM THEN 300
180 PRINT "OK." : CLOSE : END
300 PRINT "ERRORE NEI DATA !" : CLOSE : END
310 REM
1000 DATA 233,201, 0, 0, 0, 0, 85,115,111, 58, 32, 77, 79, 86
1001 DATA 88, 73, 82, 32,118,101, 99, 99,104,105,111, 45,112, 97,116
1002 DATA 104,110, 97,109,101, 32,110,117,111,118,111, 45,112, 97,116
1003 DATA 104,110, 97,109,101, 46, 36, 77, 97,110, 99, 97, 32,105,108
1004 DATA 32,110,117,111,118,111, 32,110,111,109,101, 46, 36, 13, 10
1005 DATA 70,105,108,101, 32,110,111,110, 32,116,114,111,118, 97,116
1006 DATA 111, 32,110,101,108, 32,112,101,114, 99,111,114,115,111, 32
1007 DATA 99,111,114,114,101,110,110,116,101, 46, 36, 13, 10, 00,101,114
1008 DATA 99,111,114,115,111, 32,111, 32,102,105,108,101, 32,110,111
1009 DATA 110, 32,116,114,111,115, 97,116,111, 46, 36, 13, 10, 83,111
1010 DATA 108,111, 32,115,117,108,108,111, 32,115,116,101,115,115,111
1011 DATA 32,108,101,116,116,111,114,101, 46, 36, 13, 10, 65, 99, 99
1012 DATA 101,115,115,111, 32,105,109,112,111,115,115,105, 98,105,108
1013 DATA 101, 46, 36, 13, 10, 79,107, 46, 36,190,128, 0,128, 60, 0
1014 DATA 116,105, 70,128, 60, 32,116,250,137, 54, 3, 1,128, 60, 32
1015 DATA 116, 8,128, 60,13,116, 89, 70,235,243,198, 4, 0, 70,128
1016 DATA 60, 32,116,250,128, 60,13,116, 72,137, 54, 5, 1,128, 60
1017 DATA 13,116, 3, 70,235,248,198, 4, 0,198, 68, 1, 36,139, 62
1018 DATA 3, 1,180, 9,205, 33,139, 22, 3, 1,139, 62, 5, 1,134
1019 DATA 0, 86,205, 33, 61, 2, 0,116, 37, 61, 3, 0,116, 36, 61
1020 DATA 6, 0,116, 45, 61,17, 0,116, 34,141, 22,198, 1,235, 38
1021 DATA 141, 22, 7, 1,235, 32,198, 68, 1, 36,141, 22, 52, 1,235
1022 DATA 22,141, 22, 73, 1,235, 16,141, 22,115, 1,235, 30,141, 22
1023 DATA 146, 1,235, 4,141, 22,175, 1,180, 9,205, 33,205, 32, -1
1024 DATA 30346
```

MOVDIR, si aspetta una stringa ASCII (caratteri + un byte zero finale) che rappresenta il vecchio pathname, puntata da DS:DX, e un'altra puntata da ES:DI che rappresenta il nuovo.

Se l'operazione non è fattibile il Dos, al ritorno dall'interrupt 21h, ci fa trovare nel registro AX (precisamente AL) dei codici di errore. Notare che è possibile specificare un drive diverso dal corrente in testa ai due pathname, purché nel vecchio e nel nuovo sia sempre il medesimo! Fornisco in allegato il listato MOVDIR.ASM in Assembler 8088, nonché un generatore Basic per chi non fosse in possesso dell'assemblatore.

MOVDIR.ASM è organizzata a partire dall'offset 0100h, subito dopo il Program Segment Prefix, come file COM. Questo per compattezza e semplicità. Quindi per assemblarla ad esempio col MASM Microsoft-IBM i passi sono

```
A> MASM MOVDIR;
A> LINK MOVDIR;
A> EXE2BIN MOVDIR.EXE MOVDIR.COM
A> ERASE MOVDIR.EXE
```

Osservando il semplice listato Assembler vedrete che la maggior parte del programma è dedicata all'interfaccia con l'utente.

Se dimentichiamo di passare i parametri il programma si ferma immediatamente con un messaggio esplicativo.

Inizialmente MOVDIR cerca nella linea di comando (offset 0080h e seguito) i due nomi-parametri separati da almeno uno spazio e conclusi con un ritorno carrello. Memorizza gli effettivi inizi delle due stringhe nei puntatori NAMEADDR1 e NAMEADDR2 per poi assegnarli a DS:DX e ES:DI.

Nel nostro caso ES e DS coincidono.

Poi completa i nomi con il byte zero richiesto e un ulteriore terminatore «\$» per visualizzarli successivamente (funzione 9 dell'Int 21h).

Infine chiama la funzione 56h e al ritorno controlla alcuni possibili codici di errore che ci segnala.

Purtroppo MOVDIR non accetta l'uso delle wildcard.

OFFDIR e ONDIR

Nella directory ogni nome di file è seguito da un byte detto di attributo,

```
*****
;* Felice Sobrero per MC Microcomputer - Aprile 1987 *
;* Nome della routine : MOVDIR.COM *
;* Rename esteso con cambio di percorso *
;* Uso: MOVDIR vecchio-pathname nuovo-pathname *
;* Nota: convertire con EXE2BIN in formato COM *
*****
CODE SEGMENT
ASSUME CS:CODE,DS:CODE,ES:CODE ; codice,dati,extra in questo segmento
ORG 0100h ; punto di origine per file .COM
START: JMP BEGINNING
-----
- Area di dati nel segmento di codice -
-----
COMLINE EQU 80h
RTN EQU 0Dh
SPC EQU 20h
LFD EQU 0Ah
NAMEADDR1 DW ?
NAMEADDR2 DW ?
NONAME1 DB 'Uso: MOVDIR vecchio-pathname nuovo-pathname.$'
NONAME2 DB 'Manca il nuovo nome.$'
NOFILE DB RTN,LFD,'File non trovato nel percorso corrente.$'
NOPATH DB RTN,LFD,'Percorso o file non trovato.$'
NODRIVE DB RTN,LFD,'Solo sullo stesso lettore.$'
INHIB DB RTN,LFD,'Accesso impossibile.$'
ALLOK DB RTN,LFD,'Ok.$'
-----
BEGINNING: MOV SI,COMLINE ; punta a 80h command-line
CMP BYTE PTR [SI],0 ; argomento = 0 ?
JE LBNONAME1 ; allora messaggio di errore
SPACES: SI ; punta al prossimo byte
CMP BYTE PTR [SI],SPC ; e' uno spazio ?
JE SPACES ; avanti un altro byte
MOV NAMEADDR1,SI ; se no, inizio lo nome
GOODSTRING1: CMP BYTE PTR [SI],SPC ; separatore tra due nomi ?
JE DONE1 ; lo nome finito
CMP BYTE PTR [SI],RTN ; fine linea di comando ?
JE LBNONAME2 ; errore, manca il 2o nome
INC SI ; se no, avanti un altro byte
JMP SHORT GOODSTRING1 ; ed esplora ancora
DONE1: MOV BYTE PTR [SI],0 ; zero finale stringa AsciiZ
-----
SPACES2: INC SI ; avanti un byte
CMP BYTE PTR [SI],SPC ; e' uno spazio ?
JE SPACES2 ; avanti un altro byte
CMP BYTE PTR [SI],RTN ; fine linea di comando ?
JE LBNONAME2 ; errore, manca il 2o nome
MOV NAMEADDR2,SI ; se no, inizio 2o nome
GOODSTRING2: CMP BYTE PTR [SI],RTN ; fine linea di comando ?
JE DONE2 ; 2o nome finito
INC SI ; se no, avanti un altro byte
JMP SHORT GOODSTRING2 ; ed esplora ancora
DONE2: MOV BYTE PTR [SI],0 ; zero finale stringa AsciiZ
MOV BYTE PTR [SI+1],36 ; carattere delimitatore $
-----
MOV DX,NAMEADDR1 ; puntatore a lo nome file
MOV AH,9 ; display stringa$ completa
INT 21h ; servizi vari del Dos
-----
MOV DX,NAMEADDR1 ; DS:DX puntatore a nome vecchio
MOV DI,NAMEADDR2 ; ES:DI puntatore a nome nuovo
MOV AX,5600h ; AH=86 funzione rename esteso
INT 21h ; servizi vari del Dos
-----
CMP AX,0002h ; codice di ritorno = 2
JZ LBNONFILE ; = file non trovato
CMP AX,0003h ; codice di ritorno = 3
JZ LBNOPATH ; = percorso non trovato
CMP AX,0006h ; codice di ritorno = 6
JZ LBINHIB ; = accesso inibito
CMP AX,0011h ; codice di ritorno = 17
JZ LBNODRIVE ; = non e' lo stesso lettore
LEA DX,ALLOK ; se no, operazione riuscita
JMP SHORT BYE ; visualizza e termina
LBNONAME1: LEA DX,NONAME1 ; punta al messaggio
JMP SHORT BYE ; visualizza e termina
LBNONAME2: MOV BYTE PTR [SI+1],36 ; mette un terminatore $
LEA DX,NONAME2 ; punta al messaggio
JMP SHORT BYE ; visualizza e termina
LBNONFILE: LEA DX,NOFILE ; punta al messaggio
JMP SHORT BYE ; visualizza e termina
LBNOPATH: LEA DX,NOPATH ; punta al messaggio
JMP SHORT BYE ; visualizza e termina
LBNODRIVE: LEA DX,NODRIVE ; punta al messaggio
JMP SHORT BYE ; visualizza e termina
LBINHIB: LEA DX,INHIB ; punta al messaggio
BYE: MOV AH,9 ; funzione display stringa$
INT 21h ; servizi del Dos
INT 20h ; termine programma
-----
CODE ENDS
END START
-----
; Fine
-----
```

```

*****
* Felice Sobrero per MC Microcomputer Aprile 1987 *
* Nome della routine : OFFDIR.COM *
* Fissa gli attributi Archivio / Nascondo / Solo lettura *
* di un file passato come argomento *
* Nota: da convertire con EXEBIN in formato COM *
*****
CODE SEGMENT
ASSUME CS,DS:CODE ; codice e dati in questo segmento
ORG 0100h ; punto di origine per file .COM
START: JMP BEGINNING

- Area dei dati nel segmento di codice -
-----
COMLINE EQU 80h
RTN EQU 0Dh
SPC EQU 20h
VOID EQU 0A8h
NAMEADDR DW ?
NONAME DB 'Usa: OFFDIR nome-file.$'
BADPARAM DB 'Comando errato.$'
NOFILE DB 'file non trovato.$'
NOPATH DB 'percorso non trovato.$'
INHIB DB 'accesso impossibile.$'
ALLOK DB 'file nascosto e a sola lettura.$'

BEGINNING: MOV SI,COMLINE ; punta a 80h linea di comando
CMP BYTE PTR [SI],0 ; guarda se argomento = 0
JE LBNONAME ; allora messaggio di errore
SPACES: SI ; punta al prossimo byte
INC BYTE PTR [SI],SPC ; controlla se e' uno spazio
CMP SPACES ; avanti un altro byte
JE MOV NAMEADDR,SI ; se no, memorizza il puntatore
GOODSTRING: CMP BYTE PTR [SI],RTN ; se e' un carriage return
JE LETSGO ; nome finito, siamo pronti
INC SI ; se no, avanti un altro byte
CMP SI,VOID ; se il puntatore e' a 168
JG LBSILLY ; abbandona comando troppo lungo
JMP SHORT GOODSTRING ; se no, esplora ancora
LETSGO: MOV BYTE PTR [SI],0 ; zero finale stringa AsciiZ
MOV BYTE PTR [SI+1],'$' ; carattere terminatore $
MOV DX,NAMEADDR ; puntatore a nome file
MOV AH,9 ; funzione display stringa$
INT 21h ; servizi vari del Dos

MOV DX,NAMEADDR ; puntatore a nome file
MOV CX,0023h ; 35=archivio hidden readonly
MOV AH,43h ; funz. attributi dei files
MOV AL,01h ; sub-funzione scrittura
INT 21h ; servizi vari del Dos

CMP AX,0002 ; codice di ritorno = 2
JZ LBNOFIL ; segnala file non trovato
CMP AX,0003 ; codice di ritorno = 3
JZ LBNOPATH ; segnala percorso non trovato
CMP AX,0005 ; codice di ritorno = 5
JZ LBINHIB ; segnala accesso inibito
LEA DX,ALLOK ; se no, operazione riuscita
JMP SHORT BYE ; visualizza e termina
LBNONAME: LEA DX,NONAME ; punta al messaggio
JMP SHORT BYE ; visualizza e termina
LBSILLY: LEA DX,BADPARAM ; punta al messaggio
JMP SHORT BYE ; visualizza e termina
LBNOFIL: LEA DX,NOFILE ; punta al messaggio
JMP SHORT BYE ; visualizza e termina
LBNOPATH: LEA DX,NOPATH ; punta al messaggio
JMP SHORT BYE ; visualizza e termina
LBINHIB: LEA DX,INHIB ; punta al messaggio
BYE: MOV AH,9 ; funzione display stringa$
INT 21h ; servizi vari del Dos
INT 20h ; termine programma

CODE ENDS
END START

```

OFFDIR. BAS

```

100 OPEN "offdir.com" AS #1 LEN = 1
110 FIELD #1,1 AS A$
120 CHECKSUM = 0 : RESTORE
130 READ BYTEX
140 IF BYTEX = -1 THEN 170
150 LSET A$ = CHR$(BYTEX) : PUT #1
160 CHECKSUM = CHECKSUM + BYTEX : GOTO 130
170 READ CONTROL : IF CONTROL <> CHECKSUM THEN 300
180 PRINT "OK." : CLOSE : END
300 PRINT "ERRORE NEI DATA !" : CLOSE : END
310 REM
1000 DATA 233,169, 0, 0, 0, 69,114,114,111,114,101, 32, 58, 32,109
1001 DATA 97,110, 99, 97, 32,110,111,109,101, 32,102,105,108,101, 46
1002 DATA 36, 67,111,109, 97,110,100,111, 32,101,114,114, 97,116,111
1003 DATA 46, 36, 32, 58, 32,102,105,108,101, 32,110,111,110, 32,116
1004 DATA 114,111,118, 97,116,111, 46, 36, 32, 58, 32,112,101,114, 99
1005 DATA 111,114,115,111, 32,110,111,110, 32,116,114,111,118, 97,116
1006 DATA 111, 46, 36, 32, 58, 32, 97, 99, 101,115,115,111, 32,105
1007 DATA 109,112,111,115,115,105, 98,105,108,101, 46, 36, 32, 58, 32
1008 DATA 102,105,108,101, 32,110, 97,115, 99,111,115,116,111, 32, 97
1009 DATA 32,115,111,108, 97, 32,108,101,116,116,117,114, 97, 46, 36
1010 DATA 40, 99, 41, 32, 70, 46, 83,111, 98,114,101,114,111, 32, 49
1011 DATA 57, 56, 55, 46, 13, 10, 36,232,115, 0,190,128, 0,128, 60
1012 DATA 0,116, 73, 70,128, 60, 32,116,250,137, 54, 3, 1,128, 60
1013 DATA 13,116, 9, 70,129,254,168, 0,127, 57,235,242,198, 4, 0
1014 DATA 198, 68, 1, 36,139, 22, 3, 1,180, 9,205, 33,139, 22, 3
1015 DATA 1,185, 35, 0,180, 67,176, 1,205, 33, 61, 2, 0,116, 28
1016 DATA 61, 3, 0,116, 29, 61, 5, 0,116, 30,141, 22,117, 1,235
1017 DATA 28,141, 22, 5, 1,235, 22,141, 22, 31, 1,235, 16,141, 22
1018 DATA 47, 1,235, 10,141, 22, 68, 1,235, 4,141, 22, 93, 1,180
1019 DATA 9,205, 33,205, 32,141, 22,150, 1,180, 9,205, 33,195, -1
1020 DATA 25144

```

che il comando DIR non mostra.
 I sei bit più bassi agiscono da flag con questo significato, se accesi:
 0 (1) = File a sola lettura (non si può cancellare né riscrivere).
 1 (2) = File nascosto (non compare visibilmente in directory).
 2 (4) = File di sistema (apparentemente senza alcun effetto).
 3 (8) = Etichetta di volume.
 4 (16) = Nome di subdirectory.
 5 (32) = File archivio (interessa solo BACKUP/RESTORE da disco fisso).

Le ultime versioni Dos mettono a disposizione il comando esterno ATTRIB che agisce proprio sul bit più basso per rendere un file libero o bloccato. Ma la funzione 43h sempre dell'interrupt 21h offre qualcosa di più generale. Permette di leggere o scrivere il byte di attributo a piacimento, e senza pasticciare personalmente nei settori riservati alla directory.

Come al solito si fornisce una stringa ASCIIZ puntata da DS:DX per il nome del file.

Per scrivere il byte di attributo ne metteremo il valore in CX e 1 in AL. Per leggerlo metteremo 0 in AL e ritroveremo il byte in CX al ritorno. A titolo di esempio ho preparato due routine, ONDIR e OFFDIR, strutturate in modo del tutto simile alla MOVDIR di cui ho parlato prima.

OFFDIR rende un file invisibile oltreché protetto a sola lettura.

ONDIR lo riporta libero e visibile.

Serviranno per celare dati riservati su un disco evitando di poterli cancellare inavvertitamente, o per deboli forme di protezione software.

Il bello è che OFFDIR usato con nomi di subdirectory non impedisce poi di accedervi, ed usato con nomi di file non impedisce di leggerli (con TYPE) o di lanciarli (se COM, BAT, EXE).

E se uno poi si dimentica il nome? Niente paura, CHKDSK/V li rivela.



Posseggo un Olivetti M-24 e leggo da poco la vostra rivista; vi scrivo per porvi alcune domande sperando che abbiate il tempo (e lo spazio) per rispondermi.

A) Da cosa derivano le lettere iniziali del GW-BASIC (GW) della Microsoft?

